

Постройте объектно-ориентированную модель предметной области по вариантам. Предусмотрите возможность расширения модели по новым типам объектов и поведению. Для обеспечения слабой межмодульной связанности и расширяемости программной системы примените шаблоны проектирования GoF. Необходимо использовать **не менее 2 шаблонов из каждой группы**: порождающие, структурные, поведенческие. Создайте консольное приложение на языке программирования Java, реализующее построенную объектно-ориентированную модель. Рекомендуется все проектные решения предварительно согласовывать с преподавателем.

Вариант 1

Посетитель может заказать себе напитки: чай (черный или зеленый), кофе (черный или с молоком) разных производителей. Дополнительно в чай или кофе может быть положен сахар. Каждый компонент обладает своей стоимостью. Реализовать классовую модель и приложение, позволяющее формировать меню и пересчитывать его стоимость.

Вариант 2

Предложите классовую модель вычислительной системы, собираемой из компонент. Для каждого компонента необходимо хранить идентификатор, тип, наименование, сведения о производителе, сведения о компонентной совместимости. Реализуйте приложение, которое выдает все возможные конфигурации вычислительных систем из описанных компонента

Вариант 3

Откорректируйте фабрику DocFactory:

- добавьте возможность хранения ссылок на все созданные ей объекты,
- возможность удаления ссылки на созданные ей объекты.

Создайте логгер, который реагирует на события создания объекта, изменения состояния объекта, удаления ссылки на объект и выводит в консоль следующую информацию.

- для создания (класс создаваемого объекта, начальные значения полей объекта, дата и время создания объекта)

- для изменения состояния объекта (класс создаваемого объекта, наименования изменяемых полей за последний период времени с указанием точного времени изменения)
- для удаления ссылки (класс создаваемого объекта, дата и время удаления ссылки на объект)

Логгер выводит данные с периодичностью T сек, задаваемой в его настройках.

Вариант 4

Предложите систему команд и реализуйте консольную утилиту, позволяющую создавать графовые структуры, добавлять узлы и связи, причем каждый узел определяется именем и описанием, а каждая связь — именем узла источника, именем узла приемника, описанием отношения. Полученный граф необходимо распечатать в консоли, в форматах матриц смежности и инцидентности.

Вариант 5

Реализуйте систему, условно включающую подсистему генерации данных и подсистему обработки данных. Генератор данных выполняется в цикле и с задержкой в интервале времени $[3,10]$ сек создает список из N элементов либо целого, либо строкового типа. Тип генератора настраивается в конструкторе. Подсистема обработки данных содержит

*****обработчики списков целых чисел:

- первый обработчик возвращает их сумму, среднее значение, минимальный и максимальный элемент,
- второй обработчик — удаляет из списка четные или нечетные элементы, в зависимости от настройки.

*****обработчики списков строк:

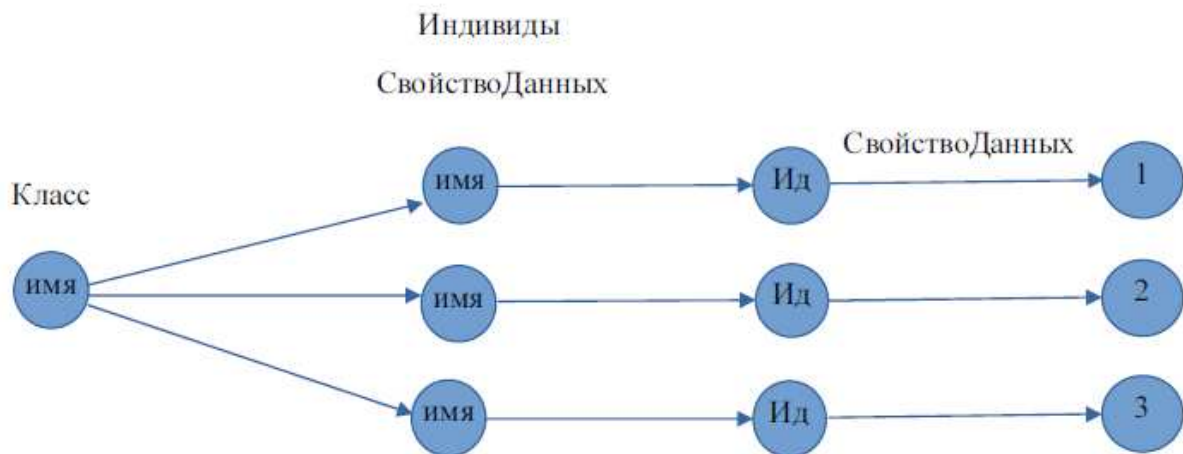
- первый обработчик — возвращает «склеенную» строку
- второй обработчик подсчитывает количество слов в каждой строке, общее количество слов и символов по всем строкам списка.

Результаты обработки передаются в общий логгер, который выводит данные в консоль.

Вариант 6

Реализуйте консольную утилиту, позволяющую создавать графовые структуры, добавлять узлы и связи, причем каждый узел определяется именем и типом, а каждая связь — именем узла источника, именем узла приемника, типом. Для узлов определены следующие типы: класс, индивид, атрибут, значение. Для связей определены следующие типы: объектное свойство, свойство данных. Приложение должно позволять:

1. Создавать одиночные узлы-классы
2. Создавать нескольких индивидов одного класса, при этом свойство данных «ИмеетИндивида» должно формироваться автоматически и для каждого индивида автоматически создается атрибут «Идентификатор» с уникальным номером в пределах всех узлов-атрибутов текущего индивида



3. Создание нескольких подклассов одного класса, при этом объектное свойство «Подкласс» должно формироваться автоматически. Полученный граф необходимо распечатать в консоли в произвольной, но понятной текстовой форме.

Вариант 7

Разработайте объектно-ориентированный каркас и программу, определяющую систему событийного управления вызовами алгоритмов на основе недетерминированного конечного автомата. Предусмотрите добавление событий и переходов между событиями с консоли. Функции переходов могут вызываться:

- а) при завершении работы отдельных алгоритмов, связанных с конкретными событиями,
- б) в результате выполнения проверки глобальных параметров, имитирующих каналы передачи данных между алгоритмами.

Вариант 8

Разработайте объектно-ориентированную модель телефона. Телефон обладает атрибутами: номер, баланс, вероятность поступления звонка. С телефоном можно выполнить следующие операции: позвонить, ответить на звонок, завершить разговор, пополнить баланс. Телефон может находиться в следующих состояниях: Ожидание, Звонок, Разговор, Заблокирован (баланс отрицательный). Реализуйте приложение, демонстрирующее переход телефона между состояниями.

Вариант 9

Реализуйте модель документа, которая обеспечивает хранение всей истории изменений значений своих атрибутов. Реализуйте программу, которая позволяет:

1. Вносить изменения в документ
2. Фиксировать изменения значений (для сохранения текущих изменений)
3. Выполнять откат состояния документа к любым изменениям, сделанным ранее Система должна быть расширяема по документам любой структуры.

Вариант 10

Требуется создать объектно-ориентированную модель контекстного меню, которое:

- а) меняет количество своих позиций в зависимости от объекта, с которым оно связывается;
- б) обладает возможностью связывания с пунктами различных алгоритмов, реализуемых внешними классами.

Вариант 11

Разработайте и реализуйте объектно-ориентированную модель фабрики с поддержкой многоуровневого кэширования объектов классов "синглетонов". Кэш-память первого уровня хранит наиболее часто запрашиваемые объекты, кэш-память второго уровня хранит объекты, которые запрашиваются реже и т.д. На последнем уровне вызываются фабричные методы, которые возвращают ссылки на запрашиваемые объекты. Максимально возможное количество объектов, хранящихся на каждом уровне, задается пользователем. Также необходимо вести учет частоты запросов объектов, на основании

которой принимается решение о кэшировании объекта, а также перемещении его между кэш-памятью разных уровней или уничтожения их кэш-памяти при снижении частоты запросов.

Вариант 12

Разработайте и реализуйте объектно-ориентированную модель, которая обеспечивает возможность:

- а) создания и инициализации объектов на основе их описания во внешних текстовых файлах, причем набор полей примитивного типа может быть произвольным,
- б) мониторинга объектов каждого типа и их количества, которые используются приложением.

Вариант 13

Разработайте и реализуйте объектно-ориентированную модель, на основе которой можно будет создать автоматизированную систему регистрации и сопровождения индивидуальных траекторий обучения. Индивидуальная траектория обучения складывается из:

- обязательных для всех дисциплин;
- рекомендуемых факультативных дисциплин, из которых учащимся может быть выбрано не менее n дисциплин.

В рамках обязательных и факультативных дисциплин могут выполняться учебные проекты, которые представляют собой множество задач, причем каждая задача может быть зарегистрирована в качестве альтернативного варианта лабораторной или практической работы.

Вариант 14

Учебный план представляет собой последовательность связанных дисциплин, причем каждая дисциплина на входе принимает и на выходе выдает три входных параметра: знания, умения, навыки (З,У,Н), которые могут быть представлены в различной форме: строка, целое значение, xml и т.д. Дисциплины могут быть связаны друг с другом, если выход (З, У, Н) предшествующей является подмножеством входа (З, У, Н) следующей дисциплины.

Разработайте и реализуйте объектно-ориентированную модель, позволяющую создавать цепочки связанных дисциплин по ЗУНам, которые рассматриваются в качестве входных и выходных интерфейсов.

Вариант 15

Реализуйте программную модель панели управления, собираемую из:

- кнопок;
- ламп цветовой индикации.

Кнопки могут находиться в двух состояниях: "нажата", "отпущена". Каждая кнопка может быть настроена на включение одной и более ламп, причем для каждой лампы задается определенный цвет индикации. Кнопки и лампы размещаются в контейнере размером NxM ячеек, в одной ячейке может быть размещена одна кнопка или одна лампа.

Доступ к каждой ячейке по координатам (x,y),

- где x - индекс, ячейки по горизонтали,
- y - индекс ячейки по вертикали.

Результаты работы программной модели представьте в текстовом формате:

1. Визуализация нажатой кнопки - "о"
2. Визуализация отпущенной кнопки - "О"
3. Визуализация неактивной лампы - "Л"
4. Визуализация активной лампы - "Л_ц", где ц - код цвета индикации

Размещение кнопок и ламп происходит случайным образом, конфигурирование кнопок и ламп выполняется при генерации панели управления случайным образом.

Пример:

Введите параметры панели:

4

2

Сгенерирована панель управления.

О - О - О - Л_

О - О - Л_ - О

Нажмите кнопку: __

1

0

Новое состояние панели

О - О - О - Л_ж

О - о - Л_з - О

Нажмите кнопку: __

Вариант 16

Разработайте и реализуйте объектно-ориентированную модель, рассчитывающую общую стоимость и вес покупки в магазине. Сама покупка собирается из различных товаров и продуктов, которые покупает отдельный человек. Известно, что все товары и продукты обладают своей стоимостью и весом. Предусмотрите возможность добавления новых общих свойств.

Вариант 17

Разработайте и реализуйте объектно-ориентированную модель ежедневника студента, с поддержкой привязки произвольных записей и мероприятий с адресом их проведения к дате и моменту или интервалу времени. Интервалы времени должны настраиваться гибко. При запуске программа должна скидывать в консоль уведомления о мероприятиях, которые скоро наступят.

Вариант 18

Разработайте объектно-ориентированную модель и программу, позволяющую собирать тест из вопросов и режимов ответа на вопрос. Необходимо обеспечить возможность добавления новых режимов ответов и правил оценивания результатов тестирования.

Вариант 19

Разработайте объектно-ориентированную модель и программу, для хранения информации о маршрутах общественного транспорта, загруженности участков (отрезки пути между остановками), а также для получения комбинированных маршрутов, с помощью которых можно добраться из пункта А в пункт В. Программа должна быть расширяема по новым маршрутам и алгоритма оценки загруженности.

Вариант 20

Разработайте объектно-ориентированную модель для конструктора расчётов рациона (количества белков, жиров, углеводов). Система должна быть расширяема по характеристикам человека, по режимам физической деятельности и ожидаемым результатам. Система должна поддерживать справочник продуктов растительного и животного происхождения. Также необходимо обеспечить возможность настройки комбинированных режимов питания с циклами различной длительности.

Вариант 21

Разработайте объектно-ориентированную модель для хранения данных о планировании, организации и проведении культурно-развлекательных мероприятий. Система должна быть расширяема по:

- типам мероприятий (свадьба, новогодний утренник, открытие торговых центров и т.д.);
- ресурсам, включая описание площадок (помещений), где планируется проведение мероприятия;
- кадровому обеспечению мероприятия (ведущий, звукооператор, фотограф и др.).

Также необходимо предусмотреть возможность формирование заказов, отмен мероприятий и формирование их расписания.